Local Feature Discriminant Projection

Mengyang Yu, *Student Member, IEEE*, Ling Shao, *Senior Member, IEEE*, Xiantong Zhen, and Xiaofei He, *Senior Member, IEEE*

Abstract—In this paper, we propose a novel subspace learning algorithm called Local Feature Discriminant Projection (LFDP) for supervised dimensionality reduction of local features. LFDP is able to efficiently seek a subspace to improve the discriminability of local features for classification. We make three novel contributions. First, the proposed LFDP is a general supervised subspace learning algorithm which provides an efficient way for dimensionality reduction of large-scale local feature descriptors. Second, we introduce the Differential Scatter Discriminant Criterion (DSDC) to the subspace learning of local feature descriptors which avoids the matrix singularity problem. Third, we propose a generalized orthogonalization method to impose on projections, leading to a more compact and less redundant subspace. Extensive experimental validation on three benchmark datasets including UIUC-Sports, Scene-15 and MIT Indoor demonstrates that the proposed LFDP outperforms other dimensionality reduction methods and achieves state-of-the-art performance for image classification.

Index Terms—Dimensionality reduction, local feature, image-to-class distance, fisher vector, image classification

1 Introduction

Recently, the use of local features has gained great popularity in computer vision. Based on local feature descriptors, e.g., SIFT [1], the sparse coding algorithm [2], dictionary learning [3], the naive Bayes nearest neighbor (NBNN) classifier [4], and Fisher kernels (FK) [5] have achieved state-of-the-art performance for image classification [6], [7]. Nevertheless, the increasingly large quantity of local feature descriptors makes local feature based algorithms severely restricted and even computationally intractable on large-scale data spaces. Dimensionality reduction algorithms [8], [9], [10], [11], [12] are needed to reduce the computational complexity. However, due to the huge number N (up to 100 M) of local feature descriptors, traditional algorithms [13], [14], e.g., manifold learning using nearest neighbor search (NN-search) with a computational complexity of at least $O(N^2)$, tend to be computationally prohabitive. Efficient algorithms are highly desirable to handle such huge amount of local feature descriptors for dimensionality reduction.

Furthermore, local feature descriptors, e.g., SIFT, are typically constructed in an unsupervised way, which would be less discriminative and contain redundant information. In contrast, supervised subspace learning [15] can not only reduce dimensions of local feature descriptors by removing redundant features but also improve the discriminability of local feature descriptors for classification. In fact, the label information could be used to achieve supervised dimensionality reduction of local feature descriptors, which however has not previously been investigated in the literature.

In this paper, we propose a novel, efficient supervised subspace learning algorithm called Local Feature Discriminant Projection (LFDP) for dimensionality reduction of local features. Most dimensionality reduction methods are performed on the image representation level, while this paper focuses on the local feature level. LFDP offers an efficient discriminant analysis which can not only reduce the dimensionality but also enhance discriminative ability of local features. To achieve a supervised local feature reduction, we adopt the image-to-class (I2C) distance [4], [10], [16] which provides an effective measurement of distances between images and classes by incorporating class label information into local features. The discriminative analysis is established by adopting the Differential Scatter Discriminant Criterion (DSDC) [17], [18] into the I2C based image representations. The advantage of using DSDC is the avoidance of the matrix singularity problem [19], a shortcoming of LDA, which enables more accurate computation. Towards efficient computation of I2C distances, we use k-means clustering to reduce the range of NN-search into the centroids of local feature clusters in each class, which makes our algorithm computational efficient without compromising the performance.

With the DSDC, we build our objective function to minimize the within-class variance while maximizing the between-class variance. However, the solution of our objective function is non-trivial due to its quartic form. We use the gradient descent algorithm on a sphere to solve this problem. In addition, an orthogonality constraint is imposed on the projections to make the subspace more compact and less redundant [8]. Unfortunately, existing orthogonalization methods [20], [21] cannot be straightforwardly applied to our scheme since they only orthogonalize the projections of the eigen-decomposition problem, which motivates us to propose a general orthogonalization on the projections via an induction method. The proposed generalized orthogonalization can also be widely applied to any other projection optimization problems. To summarize, the proposed LFDP possesses the following attractive merits:

- Unrestricted dimension: Unlike LDA, in which the reduced dimension is restricted by the number of classes, LFDP can project data onto any lower-dimensional space without suffering from the matrix singularity problem.
- $O(N)$ complexity: The time complexity of our algorithm is linear for N. In contrast to most manifold learning methods that need at least $O(N^2)$ time, our algorithm can be practically used for dimensionality reduction on large-scale data spaces.
- Generalized orthogonalization: The proposed orthogonalization method is more general and intuitive than previous methods [20], [21], and can also be applied to any other algorithms that need to compute projection matrices with the orthogonality constraints.

2 Related Work

Principal Component Analysis (PCA) is a popular dimensionality reduction method that can be directly applied to local features, which, like most unsupervised methods, makes the reduced features relatively less discriminative compared to supervised methods. Ke and Sukthankar [22] applied PCA to project the gradient image vector of a patch to a more compact descriptor, which is shorter than the standard SIFT descriptor but more robust to image deformations. Existing manifold learning algorithms, e.g., Laplacian Eigenmap (LE) [23], Locally Linear Embedding (LLE) [24] and ISOMAP [25], were proposed to learn the nonlinear structure of the data manifold. These algorithms suffer from the out-of-sample problem [26]. Locality Preserving Projections (LPP) [27] and Neighborhood Preserving Embedding (NPE) [28] as the linearized versions of LE and LLE, respectively, were developed to solve the out-of-sample problem. As unsupervised methods, they can be used for both global and local feature reduction.

- *M. Yu and L. Shao are with the Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne NE1 8ST, United Kingdom. E-mail: {m.y.yu, ling.shao}@ieee.org.*
- *X. Zhen is with the Department of Medical Biophysics, The University of Western Ontario, London, ON N6A 4V2, Canada. E-mail: zhenxt@gmail.com.*
- *X. He is with the State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang 310058, China. E-mail: xiaofeihe@cad.zju.edu.cn.*

Manuscript received 11 Sept. 2014; revised 20 Aug. 2015; accepted 28 Oct. 2015. Date of publication 3 Nov. 2015; date of current version 11 Aug. 2016.
Recommended for acceptance by L. Zelnik-Manor.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2497686

However, applying them to a large number of local features is computationally infeasible due to their high complexity. Moreover, similar to PCA, their discriminative ability is limited, as class label information is not used.

Linear Discriminant Analysis (LDA) is a conventional supervised method based on the Fisher criterion, which can also be imprudently employed for local feature reduction by using the class labels of the images from which local features are extracted. However, the large variability of local features will inevitably mislead the classifier since similar local features could be shared by images from different classes. Discriminative local descriptor learning has been explored individually in [8] and [9], both of which use the same covariance matrices of pair-wise matched and unmatched feature distances to find the linear projection. Recently, Simonyan et al. [29] proposed learning local feature descriptors using convex optimization. In fact, class labels of images are not used in the learning process, which makes the projections lose connection with classification and are therefore suboptimal. These discriminative methods [8], [29] need huge amount of ground truth with matched/unmatched pairs of local feature descriptors for training, which is not applicable in a realistic setting. Zhen et al. [10] proposed a supervised algorithm named I2C Distance Discriminative Embedding (I2CDDE) for dimensionality reduction of local features, which is specifically designed for the NBNN classifier and also computationally expensive. Furthermore, these dimension reduction methods have at least $O(N^2)$ computational complexity, which severely limits their application in large-scale data spaces.

3 Local Feature Discriminant Projection

In this section, we introduce our Local Feature Discriminant Projection algorithm before which the I2C distance is revised. With image representations based on I2C distances, we build our objective function by incorporating the DSDC for discriminant analysis of local features. To solve the objective function, we present a gradient descent optimization algorithm with a novel, generalized orthogonalization procedure.

3.1 Notations

We are given n images $X_1, \ldots, X_n$ from C classes. For the c-th class, it contains n_c samples, $c = 1, \ldots, C$. Each image X_i is represented by a set of local feature descriptors $\{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{im_i}\}$, where $\mathbf{x}_{ij} \in \mathbb{R}^D$ is the jth local feature of the ith image, $j = 1, \ldots, m_i$, $i = 1, \ldots, n$. We denote $N = \sum_{i=1}^{n} m_i$ as the total number of local feature descriptors from training images.

3.2 Image-to-Class Distance

The I2C distance introduced in the naive Bayes nearest neighbor classifier [4] represents the average of the sum of all distance squares from the local feature descriptors of an image to their corresponding nearest neighbors in each class. To be specific, the I2C distance from image X_i to class c is defined as

$$D_{X_i}^{c} = \frac{1}{m_i} \sum_{j=1}^{m_i} \|\mathbf{x}_{ij} - \mathbf{x}_{ij}^{c}\|^2,$$

where $\mathbf{x}_{ij}^{c}$ is the nearest neighbor of $\mathbf{x}_{ij}$ in class c and $\|\cdot\|$ is the L_2 norm. However, in our scheme, to reduce the complexity of NN-search in the computation of I2C distances, we first employ the K-means clustering algorithm on the set of local feature descriptors of each class [30], [31], i.e., $\bigcup_{X_i \in \text{class } c} X_i$, $c = 1, \ldots, C$. The search range is now reduced to the cluster centers, i.e., we let $\mathbf{x}^c \in \text{Centroids of } \bigcup_{X_i \in \text{class } c} X_i$ for each c.

The I2C distance is a non-parametric approximation of the log-likelihood $\log p(X_i|c) = \log \prod_{j=1}^{m_i} p(\mathbf{x}_{ij}|c)$ [4]. When using Gaussian kernel density estimation, we have

$$p(\mathbf{x}|c) = \frac{1}{L_c} \sum_{k=1}^{L_c} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_k^{(c)}\|^2\right),$$

where $\mathbf{x}$ represents an arbitrary local feature descriptor and $\mathbf{x}_1^{(c)}, \ldots, \mathbf{x}_{L_c}^{(c)}$ are the local features extracted from all the images in class c. Note that with fixed centers, diagonal covariance matrices and equal weights, the density estimation turns out to be a special case of Gaussian mixture models (GMM) used in a state-of-the-art image representation called Fisher vectors [5], [32]. If we choose the centers, covariance matrices and weights of the GMM as, for instance, all of the training local features $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, diagonal matrices and equal weights respectively, we have

$$p(\mathbf{x}|\Theta) = \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{1}{2\sigma_i^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right).$$

In this case, if the number of local features in each class (L_c) is the same, the log-likelihood of the GMM is positively related to the "average" of all the I2C distances and its gradients with respect to parameters construct a Fisher vector.

Based on I2C distances, we propose local feature discriminant projection by applying a discriminant analysis to local features for supervised dimensionality reduction. It is worthwhile to highlight that our LFDP is not restricted to the I2C distance. Other measurements, e.g., Kullback-Leibler divergence, the Hausdorff distance and the Bhattacharyya distance, could also be used to measure the relationship between images and classes. More importantly, our LFDP is a general supervised algorithm for dimension reduction which can be applied to any local feature descriptors including not only the handcrafted SIFT used in this paper, but also recent deep learning based representations [33], [34].

In addition, local features reduced by our LFDP can be fed to existing different representation methods, e.g., the bag-of-words model, sparse coding, NBNN and Fisher kernels. We use the Fisher kernels for the final image representations in order to achieve state-of-the-art performance.

3.3 Discriminant Analysis

Our goal is to seek a matrix $\mathbf{w} \in \mathbb{R}^{D \times d}$ to project the original local features $\mathbf{x}_{ij}$ with dimension D to $\mathbf{w}^T\mathbf{x}_{ij}$ in a lower-dimensional but more discriminative space $\mathbb{R}^d$. Note that after applying projection matrix $\mathbf{w}$, the nearest neighbors may change. However, for the large-scale local feature space, we approximately adopt the sum of the distances from $\mathbf{w}^T\mathbf{x}_{ij}$ to the projected nearest neighbor $\mathbf{w}^T\mathbf{x}_{ij}^c$. Denote $\Delta X_{ic} = \frac{1}{\sqrt{m_i}}[(\mathbf{x}_{i1} - \mathbf{x}_{i1}^c), \ldots, (\mathbf{x}_{im_i} - \mathbf{x}_{im_i}^c)]^T \in \mathbb{R}^{m_i \times D}$. Then the projected I2C distance becomes

$$\begin{aligned}
\widehat{D}_{X_i}^{c} &= \frac{1}{m_i} \sum_{j=1}^{m_i} \|\mathbf{w}^T\mathbf{x}_{ij} - (\mathbf{w}^T\mathbf{x}_{ij})^c\|^2 \\
&\approx \frac{1}{m_i} \sum_{j=1}^{m_i} \|\mathbf{w}^T\mathbf{x}_{ij} - \mathbf{w}^T\mathbf{x}_{ij}^c\|^2 \\
&= \operatorname{tr}\big((\Delta X_{ic}\mathbf{w})(\Delta X_{ic}\mathbf{w})^T\big) \\
&= \operatorname{tr}\big((\Delta X_{ic}\mathbf{w})^T(\Delta X_{ic}\mathbf{w})\big) \\
&= \operatorname{tr}\big(\mathbf{w}^T\Delta X_{ic}^T\Delta X_{ic}\mathbf{w}\big).
\end{aligned}$$

Without loss of generality, we first consider the condition that $\mathbf{w}$ is a column vector in the algorithm, i.e., $d = 1$. In fact, we will compute the column vectors of the projection matrix one by one. In this case, the projected I2C distances of an image will be

$$\begin{aligned}\mathbf{d}_i &= (\widehat{D}^1_{X_i}, \ldots, \widehat{D}^C_{X_i}) \\ &= (\mathbf{w}^T \Delta X_{i1}^T \Delta X_{i1} \mathbf{w}, \ldots, \mathbf{w}^T \Delta X_{iC}^T \Delta X_{iC} \mathbf{w}),\end{aligned} \tag{1}$$

which is called an *I2C vector*. In other words, for each image X_i, we have a corresponding vector $\mathbf{d}_i$ in linear space $\mathbb{R}^C$ which is called *I2C vector space*. Then we have the mean of the vectors in class i and the mean of all the vectors, denoted by μ_i and μ, respectively. Having the representations with I2C vectors, we incorporate the Differential Scatter Discriminant Criterion in the I2C vector space to obtain our objective function in the following form that needs to be maximized:

$$J = \sum_{c=1}^{C} n_c \|\mu_c - \mu\|^2 - \lambda \sum_{c=1}^{C} \sum_{\mathbf{d}_k \in \text{class } c} \|\mathbf{d}_k - \mu_c\|^2, \tag{2}$$

where λ is a tuning parameter. μ_c and μ are computed by the following equations

$$\mu_c = \frac{1}{n_c} \sum_{\mathbf{d}_k \in \text{class } c} \mathbf{d}_k := (\mathbf{w}^T M_{c1} \mathbf{w}, \ldots, \mathbf{w}^T M_{cC} \mathbf{w}),$$

$$\mu = \frac{1}{N} \sum_{k=1}^{N} \mathbf{d}_k := (\mathbf{w}^T M_1 \mathbf{w}, \ldots, \mathbf{w}^T M_C \mathbf{w}),$$

where

$$M_{cj} = \frac{1}{n_c} \sum_{\mathbf{d}_k \in \text{class } c} \Delta X_{kj}^T \Delta X_{kj}, \ c, j = 1, \ldots, C,$$

and

$$M_j = \frac{1}{N} \sum_{i=1}^{N} \Delta X_{ij}^T \Delta X_{ij}, \ j = 1, \ldots, C.$$

Now we can formulate J as a function of $\mathbf{w}$ as follows:

$$\begin{aligned}J(\mathbf{w}) = &\sum_{c=1}^{C} n_c \sum_{j=1}^{C} (\mathbf{w}^T \Delta M_{cj} \mathbf{w})^2 \\ &- \lambda \sum_{c=1}^{C} \sum_{\mathbf{d}_k \in \text{class } c} \sum_{j=1}^{C} (\mathbf{w}^T V_{kj}^c \mathbf{w})^2,\end{aligned} \tag{3}$$

where $\Delta M_{cj} = M_{cj} - M_j$ and $V_{kj}^c = \Delta X_{kj}^T \Delta X_{kj} - M_{cj}$ for $\mathbf{d}_k \in$ class c, $c, j = 1, \ldots, C$.

3.4 Gradient Descent on Sphere

The classic eigen-decomposition of a matrix is not applicable to our problem due to the quartic form of the objective function. We adopt a procedure of gradient descent on a sphere to find the projection vector. Our goal is to find the optimal $\mathbf{w}$ by maximizing $J(\mathbf{w})$. To obtain the final orthonormal projection matrix, we set a norm constraint $\|\mathbf{w}\| = 1$ for each vector. However, the update rule of the traditional gradient descent for a maximization problem: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \gamma \nabla J(\mathbf{w}^{(t)})$ does not guarantee this constraint. Thus we amend the original algorithm and constrain it on the D-dimensional unit sphere.

Define two matrix-valued functions:

$$M(\mathbf{w}) = \sum_{c=1}^{C} n_c \sum_{j=1}^{C} \mathbf{w}^T \Delta M_{cj} \mathbf{w} \cdot \Delta M_{cj} \tag{4}$$

and

$$V(\mathbf{w}) = \sum_{c=1}^{C} \sum_{\mathbf{d}_k \in \text{class } c} \sum_{j=1}^{C} \mathbf{w}^T V_{kj}^c \mathbf{w} \cdot V_{kj}^c. \tag{5}$$

We obtain the gradient of $J(\mathbf{w})$:

$$\nabla J(\mathbf{w}) = 2M(\mathbf{w})\mathbf{w} - 2\lambda V(\mathbf{w})\mathbf{w}. \tag{6}$$

We project $\nabla J(\mathbf{w})$ onto the tangent direction of $\mathbf{w}$ on the sphere as $\mathbf{p} = \nabla J(\mathbf{w}) - \langle \nabla J(\mathbf{w}), \mathbf{w} \rangle \mathbf{w}$ and normalize it as $\mathbf{p}_0 = \mathbf{p}/\|\mathbf{p}\|$. By using the first-order Taylor expansion, we know $\nabla J(\mathbf{w})$ is the steepest increasing direction. For direction $\mathbf{p}$, we have $\langle \mathbf{p}, \nabla J(\mathbf{w}) \rangle = \langle \nabla J(\mathbf{w}), \nabla J(\mathbf{w}) \rangle - \langle \nabla J(\mathbf{w}), \mathbf{w} \rangle^2 = \|\nabla J(\mathbf{w})\|^2 - \|\nabla J(\mathbf{w})\|^2 \cos^2 \alpha \geq 0$, where α is the angle between $\nabla J(\mathbf{w})$ and $\mathbf{w}$. Thus $\mathbf{p}$ is still an increasing direction. Then for the tth step, we have the following update rule:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} \cos\theta + \mathbf{p}_0^{(t)} \sin\theta, \tag{7}$$

where $\theta \in [0, \pi/2]$ is the step size. Since $\mathbf{w}$ and $\mathbf{p}_0$ are orthogonal, the norm of the updated variable remains of unit length. In addition, to accelerate the convergence, we also employ an adaptive step size θ_t, i.e., if $J(\mathbf{w}^{(t+1)}) \geq J(\mathbf{w}^{(t)})$, we set $\theta_{t+1} = \min(2\theta_t, \pi/2)$, otherwise, $\theta_{t+1} = \theta_t/2$. The iterative procedure is described in Algorithm 1.

Algorithm 1. The Gradient Descent for Local Feature Discriminant Projection

Input: The local feature descriptors $\{\mathbf{x}_{ij}\}$ of each image and the parameter K in K-means.
Output: The projection vector $\mathbf{w}$ in the first dimension.
Employ K-means algorithm for the local feature set of each class;
Find the nearest neighbor $\mathbf{x}_{ij}^c$ of $\{\mathbf{x}_{ij}\}$ in the centroids of each class;
Compute matrix-valued functions $M(\mathbf{w})$ and $V(\mathbf{w})$ in Eqs. (4) and (5);
Initialize step size $\theta_1 \in (0, \pi/2)$ and randomly initialize unit vector $\mathbf{w}^{(1)}$;
repeat
  Compute the projection of $\nabla J(\mathbf{w}^{(t)})$ on the tangent direction of $\mathbf{w}^{(t)}$: $\mathbf{p}^{(t)} = \nabla J(\mathbf{w}^{(t)}) - \langle \nabla J(\mathbf{w}^{(t)}), \mathbf{w}^{(t)} \rangle \mathbf{w}^{(t)}$ and apply normalization $\mathbf{p}_0^{(t)} = \mathbf{p}^{(t)}/\|\mathbf{p}^{(t)}\|$;
  Compute $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} \cos\theta_t + \mathbf{p}_0^{(t)} \sin\theta_t$;
  repeat
    $\theta_t \leftarrow \theta_t/2$;
  until $J(\mathbf{w}^{(t+1)}) \geq J(\mathbf{w}^{(t)})$
  Update $\theta_{t+1} = \min(2\theta_t, \ \pi/2)$;
until convergence.

3.5 Orthogonality Constraints

Until now we have only computed the projection vector for the first dimension. In this section, we use the induction method to compute the remaining vectors successively and make them mutually orthogonal by using the matrix composed by previous output vectors. Previous works [8], [20] have highlighted the benefits of orthogonality constraints, for instance, avoidance of overfitting and redundancy in representing the subspace. With this orthogonalization procedure, we can establish our whole algorithm.

Suppose we have obtained the first p $(p \geq 1)$ discriminant vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots . \mathbf{w}_p$. We want to compute the next vector $\mathbf{w}_{p+1}$ to maximize $J(\mathbf{w})$ with the orthogonal constraints

$$\mathbf{w}_1^T \mathbf{w}_{p+1} = \mathbf{w}_2^T \mathbf{w}_{p+1} = \cdots = \mathbf{w}_p^T \mathbf{w}_{p+1} = 0, \tag{8}$$

and an additional norm constraint on $\mathbf{w}_{p+1}$, i.e., $\|\mathbf{w}_{p+1}\| = 1$. The method in [20] can not be applied in our scheme due to the high degree of Lagrangian in our case. We use an alternative but more

general method by basis transformation to solve this issue. In other words, we compute the next discriminant vector in a special subspace in which the orthogonal constraints vanish.

According to the inductive assumption, vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p$ should be an orthonormal basis of a subspace in $\mathbb{R}^D$. Let us denote $\text{span}V_p = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p)$ and $W_p = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p]$. Then V_p is a p-dimensional subspace and W_p is a $D \times p$ matrix. Recall that our primary goal is to seek an optimal $\mathbf{w}$ by maximizing $J(\mathbf{w})$:

$$\arg\max_{\mathbf{w}\in\mathbb{R}^D} J(\mathbf{w}). \tag{9}$$

Once we have obtained subspace V_p, $\mathbf{w}_{p+1}$ is required to be orthogonal to all the vectors in V_p. Consequently, we need to compute the constrained optimization problem

$$\arg max_{\mathbf{w}\in V_p^{\perp}} J(\mathbf{w}) \tag{10}$$

to find the solution of $\mathbf{w}_{p+1}$, where $V_p^{\perp}$ is the null space of V_p and $dim V_p^{\perp} = D - p$. Straightforwardly, the data can be projected onto subspace $V_p^{\perp}$ so that the computation process is completely performed in a $(D-p)$-dimensional linear subspace, i.e., the new coordinates are in $\mathbb{R}^{D-p}$. Then the output will be orthogonal to any vectors in V_p. For this reason, we need to find a basis $B_p = [\mathbf{b}_1, \ldots, \mathbf{b}_{D-p}] \in \mathbb{R}^{D\times(D-p)}$ of $V_p^{\perp}$. In fact, we need only to solve the linear equation $W_p^T X = 0$, which is commonly used in linear algebra. Furthermore, we make this basis orthonormal by following the Gram-Schmidt procedure.

Now with this orthonormal basis B_p, we project data from $\mathbb{R}^D$ onto subspace $V_p^{\perp}$. Specifically, for a local feature and an I2C vector, we have transformations $\mathbf{x}_{ij} \leftarrow B_p^T \mathbf{x}_{ij}$ and $\mathbf{d}_i \leftarrow (\mathbf{v}^T B_p^T \Delta X_{i1}^T \Delta X_{i1} B_p \mathbf{v}, \ldots, \mathbf{v}^T B_p^T \Delta X_{iC}^T \Delta X_{iC} B_p \mathbf{v})$, respectively, where $\mathbf{v}$ is a vector in $\mathbb{R}^{D-p}$. Then we only need to solve the unconstrained problem in a lower-dimensional space:

$$\arg\max_{\mathbf{v}\in\mathbb{R}^{D-p}} J_p(\mathbf{v}) = \arg\max_{\mathbf{v}\in\mathbb{R}^{D-p}} \left(\mathbf{v}^T M_p(\mathbf{v})\mathbf{v} - \lambda \mathbf{v}^T V_p(\mathbf{v})\mathbf{v}\right), \tag{11}$$

where $M_p(\cdot)$ and $V_p(\cdot)$ are the images of matrix-valued functions $M(\cdot)$ and $V(\cdot)$ after the projection, respectively, i.e., $\Delta M_{cj} \leftarrow B_p^T \Delta M_{cj} B_p$ and $V_{kj}^c \leftarrow B_p^T V_{kj}^c B_p$. Now it is an optimization problem where the constraints vanish and here we return to our first goal in the $(D-p)$-dimensional space.

Algorithm 2. Local Feature Discriminant Projection

Input: The input of Algorithm 1 and the target dimension d.
Output: The projection matrix $\mathbf{w}$.
Initialization: $\mathbf{w} \leftarrow \emptyset$ and $B \leftarrow I$;
for $i = 1$ to d **do**
Project training data onto the null space of span($\mathbf{w}$) by using the basis B;
Call Algorithm 1 to compute the corresponding projection vector $\mathbf{w}_i$ in subspace span$(\mathbf{w})^{\perp}$ and update $\mathbf{w}_i \leftarrow B\mathbf{w}_i$;
Update $\mathbf{w} \leftarrow [\mathbf{w}, \mathbf{w}_i]$ and let B be an orthonormal basis of span$(\mathbf{w})^{\perp}$ by solving the corresponding linear equation and following the Gram-Schmidt procedure.
end for

Having the solution $\mathbf{v}^*$ for the optimization problem (11) in $\mathbb{R}^{D-p}$, we transform it to an element in $V_p^{\perp} \in \mathbb{R}^D$. Actually, $\mathbb{R}^{D-p}$ and $V_p^{\perp}$ are two isomorphic linear spaces and B_p can be regarded as a linear isomorphism between them. Through the representation of an orthonormal basis, for each $\mathbf{w} \in V_p^{\perp}$, we have $\mathbf{w} = \sum_{i=1}^{D-p} w_i \mathbf{b}_i$, where $w_i \in \mathbb{R}$, and the inner product of $\mathbf{w}$ and $\mathbf{b}_i$ will be $\langle \mathbf{w}, \mathbf{b}_i \rangle = w_i, \forall i$. Then $(w_1, \ldots, w_{D-p})^T = (\langle \mathbf{w}, \mathbf{b}_1 \rangle, \ldots, \langle \mathbf{w}, \mathbf{b}_{D-p} \rangle)^T = [\mathbf{b}_1, \ldots, \mathbf{b}_{D-p}]^T \mathbf{w} = B_p^T \mathbf{w}$, i.e., the result of multiplying the left side of $\mathbf{w}$ by B_p^T is the coefficient of the representation by B_p. Finally, we set $\mathbf{w}_{p+1} = B_p \cdot \mathbf{v}^* \in V_p^{\perp}$ as a linear combination of B_p. The whole LFDP algorithm is illustrated in Algorithm 2.

Remark. The proposed orthogonalization procedure is a more general way to compute orthogonal projection matrices. Note that, in Algorithm 2, given the input of Algorithm 1, we need only Algorithm 1 to output a projection vector without need to know the computation process. Therefore, Algorithm 1 could be seen as a *black box* that is able to compute the projection vector (for those that output a matrix, we only need its first column). Now we have the following general proposition.

Proposition. *Given maximizing (minimizing) algorithm $\mathcal{A}$ which takes projected data $\mathbf{w}^T\mathbf{x}$ as input and outputs the optimal vector, and an orthonormal basis B_p of $(D-p)$-dimensional subspace $V_p^{\perp} \subseteq \mathbb{R}^D$, if $\mathbf{v}^*$ is the optimal solution of $\mathcal{A}(\mathbf{v}^T B_p^T \mathbf{x})$ in $\mathbb{R}^{D-p}$, $\mathbf{w}^* = B_p \mathbf{v}^*$ is the optimal solution of $\mathcal{A}(\mathbf{w}^T\mathbf{x})$ in $V_p^{\perp}$.*

3.6 Relations between Algorithm 2 and the Ordinary Eigen-Decomposition

In fact, assuming that the optimization problem is simplified to the eigen-decomposition of a symmetric matrix $A \in \mathbb{R}^{D\times D}$ such as PCA, we prove that the proposed orthogonalization method finds the same eigenvectors with the eigen-decomposition by adopting mathematical induction. Suppose $A = \sum_{i=1}^{D} \lambda_i \mathbf{w}_i \mathbf{w}_i^T = W \Lambda W^T$ is the spectral decomposition of A and $\lambda_1 \geq \ldots \geq \lambda_D$, where $\Lambda = diag(\lambda_1, \ldots, \lambda_D)$ and $W = [\mathbf{w}_1, \ldots, \mathbf{w}_D]$. Then $\mathbf{w}_i^T \mathbf{w}_j = 0$ if $i \neq j$ and $\mathbf{w}_i^T \mathbf{w}_i = 1$ for $i = 1, \ldots, D$.

For the first vector, both Algorithm 2 and the eigen-decomposition output the eigenvector $\mathbf{w}_1$ corresponding to the largest eigenvalue of A. Assume Algorithm 2 has output the first k eigenvectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$. For the $(k+1)$-th vector, $\mathbf{w}_{k+1}$ is the eigenvector corresponding to the eigenvalue λ_{k+1}. Algorithm 2 first finds an orthonomal basis $B \in \mathbb{R}^{D\times(D-k)}$ of $span(\mathbf{w}_1, \ldots, \mathbf{w}_k)^{\perp}$. Since W is an orthogonal matrix, we have $span(\mathbf{w}_1, \ldots, \mathbf{w}_k)^{\perp} = span(\mathbf{w}_{k+1}, \ldots, \mathbf{w}_D)$. Then there exists an orthogonal matrix $P \in \mathbb{R}^{(D-k)\times(D-k)}$ such that $B = W_{k+1} P$, where $W_{k+1} = [\mathbf{w}_{k+1}, \ldots, \mathbf{w}_D]$. In the $(k+1)$th step of Algorithm 2, we eigen-decompose the matrix $B^T A B$ to compute its largest eigenvalue. Note that

$$\begin{aligned} B^T A B &= P^T W_{k+1}^T \Big(\sum\nolimits_{i=1}^{D} \lambda_i \mathbf{w}_i \mathbf{w}_i^T\Big) W_{k+1} P \\ &= P^T W_{k+1}^T \Big(\sum\nolimits_{i=k+1}^{D} \lambda_i \mathbf{w}_i \mathbf{w}_i^T\Big) W_{k+1} P \\ &= P^T W_{k+1}^T W_{k+1} \Lambda_{k+1} W_{k+1}^T W_{k+1} P \\ &= P^T \Lambda_{k+1} P, \end{aligned}$$

where $\Lambda_{k+1} = diag(\lambda_{k+1}, \ldots, \lambda_D)$. Therefore, the largest eigenvalue of $B^T A B$ is still λ_{k+1}, which indicates that the corresponding eigenvalues of the output vectors of Algorithm 2 are $\lambda_1, \ldots, \lambda_D$. Then the whole output set of Algorithm 2 is $\{\mathbf{w}_1, \ldots, \mathbf{w}_D\}$ up to sign.

3.7 Complexity Analysis

Our LFDP is computationally more efficient than most of the existing manifold learning methods. We provide a complexity analysis on the two procedures: gradient descent and orthogonalization of our LFDP in terms of time complexity and memory cost, since in the test phase, the complexity depends on the classifier and the time complexity will apparently be reduced after dimensionality reduction.

TABLE 1
Comparing the Complexity of LFDP with Other Linear Algorithms on N Where K Is the Parameter of K-Means and k Is the Parameter of the KNN Algorithm

Method	LFDP	PCA	LDA	I2CDDE [10]	LDE [8]	LDP [9]	LPP [27]	NPE [28]
Complexity	$O(KN)$	$O(N)$	$O(N)$	$O(N^2)$	$O(N^2)$	$O(N^2)$	$O(kN^2)$	$O(kN^2)$

Gradient descent. During the iterative procedure of gradient descent, the main cost is induced by the computation of the I2C distances. The time complexity of a brute-force method of NN-search in K centroids with D-dimension is $O(KND)$. Computing $M(\mathbf{w})$ and $V(\mathbf{w})$ needs $O(D^2C^2)$ and $O(D^2Cn)$ time respectively, where n is the number of training images. Then the time complexity of the gradient descent with N_{iter} steps in a D-dimensional space is $O(N_{iter}(D^2C^2 + D^2Cn))$ and the time complexity of the whole procedure is at most $O(KND + N_{iter}D^2C^2)$. The memory cost of the iterative procedure is $O(D^2C^2 + D^2Cn)$.

Orthogonalization. We can observe that the main step in the orthogonalization procedure is the Gram-Schmidt procedure, which requires at most $O(nm^2)$ time and $O(nm + m^2)$ memory for computing on m n-dimensional vectors [35]. Notice that, in our algorithm, m varies from 1 to d and n varies from D to $D - d + 1$, where d is the dimension of the projected space.

In total, with the complexity $O(TKND)$ in the K-means, where T is the number of iterations in the K-means, our LFDP algorithm requires at most $O((T+1)\ KND + dN_{iter}(D^2C^2 + D^2Cn) + \frac{1}{6}d^3D)$ time complexity and $O(D^2C^2 + D^2Cn + \frac{1}{2}d^2D + \frac{1}{6}d^3)$ memory. Due to the large number of local feature descriptors, generally $N \gg D$, we show the computational complexity on N through comparing our algorithm with other dimensionality reduction methods in Table 1, where K is the parameter of K-means and k is the parameter of the k-nearest neighbor (KNN) algorithm. In fact, KNN-based algorithms highly rely on the neighborhood structure of each point, which will be changed by K-means clustering. In addition, K-means may also change the order of I2C distances where there are similar classes or noisy data points, and therefore, mislead the learning of I2CDDE leading to the failure of NBNN. In contrast, our discriminant analysis considers the relationships of intra-class and inter-class variations among I2C vectors, achieving a global optimization objective. Therefore, using K-means centroids can not only make our LFDP computationally more efficient but also tolerant to the fluctuation of I2C distances.

4 Experiments

We have extensively validated our LFDP algorithm on three widely used benchmark datasets, i.e., UIUC-Sports, Scene-15 and MIT Indoor. Experimental results show that our LFDP largely outperforms representative dimension reduction algorithms and achieves state-of-the-art performance.

4.1 Implementation Details

The optimal tuning parameter λ for each dataset is selected from one of $\{0.1, 0.2, \ldots, 1\}$, which yields the best performance by 10-fold cross-validation on the training data. We fix $K = 300$ in K-means for all datasets and set the maximum number of the K-means iteration as 20. In addition, the K-means clustering for each class can be performed in a parallel way to save time complexity. We take the Improved Fisher Kernel (IFK), which is an improved version of Fisher kernels [36], based on raw SIFT descriptors without dimension reduction as the baseline. We compare with PCA as a representative unsupervised algorithm which has shown competitive and even better performance than manifold learning algorithms including ISOMAP, LLE and LE on diverse tasks [37]. LDA is included for comparison as a supervised algorithm. The parameter k of the KNN algorithm in LPP and NPE is tuned by selecting from $\{5, 6, \ldots, 15\}$. By following the setting in [9], we randomly select 1.5×10^5 local features from all the training sets for training the projection of LDP. ISOMAP is not involved in the comparison due to the out-of-sample problem. All the experiments are implemented using Matlab 2013b on a workstation configured with an i7 processor and 32 GB RAM.

TABLE 2
Resource Requirements of Different Methods for the 900,000 SIFT Features from the UIUC-Sports Dataset

Method	Memory cost	Runtime
LFDP	1 GB	30 mins
I2CDDE	1 GB	8 hrs
LDE / LDP	1 GB	8 hrs
LPP / NPE	900 GB	16 hrs

4.2 Datasets

UIUC-Sports. The Sports event dataset was introduced in [38], consisting of eight sports event categories. The number of images in each class ranges from 137 to 250. We follow the experimental setting in [38] to randomly select 70 and 60 images per class for training and testing respectively. The procedure is repeated five times and the average is reported as the final result. Differently, we use the original images rather than the resized ones.

Scene-15. The Scene-15 dataset [39] consists of 4,485 images which are labeled in 15 distinct classes. The number of images in each class varies from 200 to 400. Following the experimental setting in [39], we randomly select 100 images in each class as training data and test the remaining images. The procedure is repeated five times and the average is reported as the final result.

MIT Indoor. The MIT Indoor scene dataset [40] contains 67 indoor scene categories for a total of 15,620 images. The number of images in each class varies from 100 to 734. Eighty and 20 images are selected in each category for training and testing respectively by following the experimental setting in [40] and the average is reported.

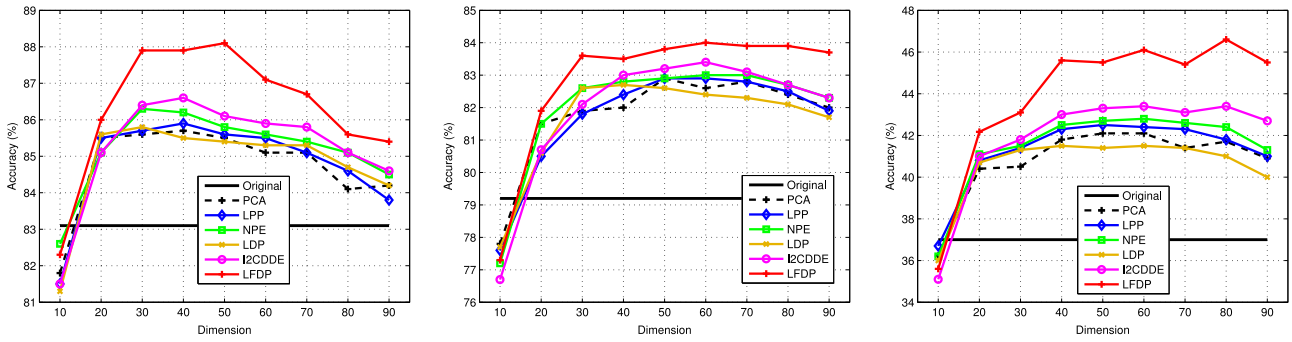
4.3 Local Feature and Classifier

We use the software provided by Yang et al. [41] to compute the SIFT descriptors. In contrast to existing works which either use multi-scale SIFT descriptors [42], spatial pyramid representation [43] or multiple descriptors [4], [42], we simply use single-scale SIFT descriptors in patches of 16×16. In our experiments, the average numbers of local features extracted from each image in three datasets are all 1,500. Then the total numbers (N) of the training local features in the above three datasets are 900,000, 2,000,000 and 8,000,000, respectively.

We employ a linear SVM classifier with IFK [36] and compute the Fisher vector for each image based on its local features by following the settings in [36] using 256 Gaussians in the GMM.

4.4 Resource Requirements

In Table 2, we list the resource requirements for training the projections by different dimensionality reduction methods. The nearest neighbor search and the computation for pairwise distances make $O(N^2)$ methods suffer from the high computational complexity.



(a) UIUC-Sports (b) Scene-15 (c) MIT Indoor

Fig. 1. Performance (percent) of linear SVMs with IFK in different lower-dimensional subspaces on the UIUC-Sports, Scene-15 and MIT Indoor datasets. Note that we only use one type of local descriptor: SIFT in single-scale patches.

Note that the runtime for LPP and NPE is a theoretical value since it is infeasible to implement them with such large memory. Therefore, to use the largest possible number of features that can be handled by our workstation, a subset consisting 1.5×10^5 local features is randomly selected from the whole training set for evaluating these methods.

4.5 Results

The performance comparison of LFDP with other dimensionality reduction methods is shown in Figs. 1a, 1b and 1c for UIUC-Sports, Scene-15 and MIT Indoor, respectively. The baseline represents the performance of SVMs with IFK in the original 128-dimensional SIFT space without dimensionality reduction. The proposed method shows consistent advantages on all the three datasets. Our method improves the baseline phenomenally with a large margin. PCA usually reaches its highest accuracy around the dimension of 50 and remains stable with the increase of dimensionality. Other methods such as LPP, NPE, LDP and I2CDDE only sightly outperform PCA. In contrast with the above methods, we can observe that LFDP goes up rapidly with the increase of the dimension when the dimension is low and achieves the competitive results around the dimension of 40 (even at 30). With the reduced local feature descriptors by LFDP, the dimensionality of Fisher vectors is several times shorter than the original dimension, which reduces the computational cost for classification but strengthens the discriminative ability due to the supervised learning.

Furthermore, the advantage of our method has been also shown by comparing with LDA. Note that LDA learns the projection matrix by directly labeling the local features with class labels of images they belong to. Since the performance of LDA is also restricted by the number of classes [44], the upper bound of reduced dimensionality of LDA is $C-1$, on which LDA reaches its best performance. We report the best results of PCA and LDA on different datasets for the comparison with the results of LFDP in Table 3. LDA with the Fisher criterion produces results below the baseline on the UIUC-Sports dataset since it contains only eight classes so that the result is obtained by seven-dimensional local descriptors. To alleviate the dimension restriction of LDA with the Fisher criterion, we implement LDA with the DSDC criterion using the parameter λ similar to Eq. (2). We tune λ in $\{0.1, 0.2, \cdots, 1\}$ and the best results are reported in Table 3. With the DSDC, the reduced dimension of LDA is not restricted by the number of classes and the results are significantly improved.

LFDP can efficiently find lower-dimensional but more discriminative feature space and achieves the state-of-the-art results [42], [45], [46], which reveals its capability in dimensionality reduction of ubiquitous local feature spaces in large scale.

TABLE 3
Performance (Percent) of Linear SVMs with IFK After PCA, LDA and LFDP Reduction on Local Features

Method	UIUC-Sports	Scene-15	MIT Indoor
Baseline	83.1 ± 0.3	79.2 ± 0.2	37.0 ± 0.3
PCA	85.7 ± 0.2	82.9 ± 0.4	42.1 ± 0.4
LDA^1	81.2 ± 0.4	79.9 ± 0.4	38.6 ± 0.5
LDA^2	85.4 ± 0.4	83.0 ± 0.3	42.3 ± 0.4
LFDP	$\mathbf{88.1 \pm 0.5}$	$\mathbf{84.0 \pm 0.5}$	$\mathbf{46.6 \pm 0.4}$

LDA^1 is the LDA with the Fisher criterion and LDA^2 is the LDA with the DSDC. The results listed in the table are their best accuracies. The baseline is the classification result of IFK without dimensionality reduction of local feature descriptors.

4.6 Algorithm Analysis

We also evaluate the performance of Algorithm 1 in terms of convergency. We randomly initialize $\mathbf{w}$ 50 times on the UIUC-Sports dataset and the average value of the objective function in Eq. (3) and the average difference $\|\mathbf{w}^{(t)} - \mathbf{w}^{(t-1)}\|$ on the first dimension are reported in Fig. 2, where t is the number of iteration and λ is fixed at 0.1. We can observe that $\mathbf{w}$ converges within only 10 steps. Therefore, we always fix the maximum number of iteration at 10 in the experiments.

In addition, LFDP achieves the best performance with a small value of K in K-means, which guarantees the computational efficiency. We have investigated the performance under different values of parameter K as shown in Table 4. On all the three datasets, our method yields the best results with $K = 300$ which is much smaller than the number of local feature descriptors, which is up to $120{,}000$ in each class. This largely reduces the computational complexity.

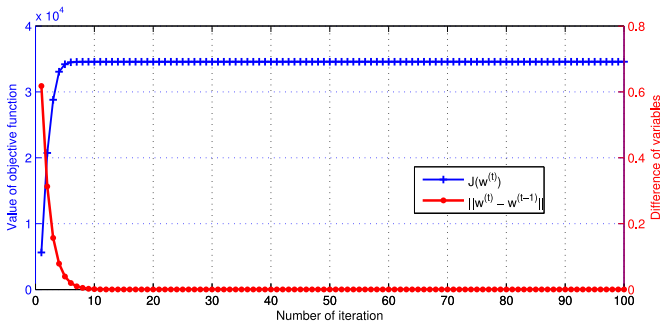


Fig. 2. The convergency of the objective function and the difference of variables with respect to the number of iteration.

TABLE 4
Comparing the Results (Percent) of LFDP with Different K Values

Dataset \ K	50	100	200	300	400	500
UIUC-Sports	76.5	83.2	86.7	**88.1**	88.0	88.0
Scene-15	75.3	82.7	83.6	**84.0**	83.8	84.0
MIT Indoor	36.7	40.3	44.8	**46.6**	46.4	46.4

The best results while varying the target dimension are listed.

5 Conclusion

A new subspace learning algorithm called Local Feature Discriminant Projection has been proposed for supervised dimensionality reduction of local features. The projections for reduction are obtained by optimizing an objective function constructed based on the Differential Scatter Discriminant Criterion and the I2C representations. A general orthogonalization method has been proposed to learn the projections which guarantees a more compact space with less redundancy. The proposed LFDP has a much lower complexity than popular manifold learning methods, providing an alternative way to efficiently analyze large-scale data. The experimental results on three widely used benchmarks for image classification have validated the effectiveness of LFDP and shown its advantages over traditional dimensionality reduction algorithms. In future work, we aim to extend our algorithm to the semi-supervised and unsupervised settings for more practical applications.

Acknowledgments

This work was supported in part by Northumbria University and in part by National Natural Science Foundation of China under Grant 61528106. The corresponding author is Ling Shao.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[2] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2006, pp. 801–808.
[3] F. Zhu and L. Shao, "Weakly-supervised cross-domain dictionary learning for visual recognition," *Int. J. Comput. Vis.*, vol. 109, nos. 1-2, pp. 42–59, 2014.
[4] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[5] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 1999, pp. 487–493.
[6] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1359–1371, Jul. 2014.
[7] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.
[8] G. Hua, M. Brown, and S. A. J. Winder, "Discriminant embedding for local image descriptors," in *Proc. IEEE Int Conf. Comput. Vis.*, 2007, pp. 1–8.
[9] H. Cai, K. Mikolajczyk, and J. Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 338–352, Feb. 2011.
[10] X. Zhen, L. Shao, and F. Zheng, "Discriminative embedding via image-to-class distances," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–12.
[11] B. Geng, D. Tao, C. Xu, L. Yang, and X. Hua, "Ensemble manifold regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1227–1233, Jun. 2012.
[12] L. Shao, L. Liu, and M. Yu, "Kernelized multiview projection for robust action recognition," *Int. J. Comput. Vis.*, pp. 1–15, 2015, Doi: 10.1007/s11263-015-0861-6.
[13] M. Sugiyama, "Local fisher discriminant analysis for supervised dimensionality reduction," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 905–912.
[14] Z. Wang, Y. Hu, and L. Chia, "Image-to-class distance metric learning for image classification," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 706–719.
[15] X. Zhen, Z. Wang, M. Yu, and S. Li, "Supervised descriptor learning for multi-output regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1211-1218.
[16] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3241–3253, Aug. 2014.
[17] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Waltham, Ma, USA: Academic Press, 1990.
[18] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[19] T. Zhang, D. Tao, and J. Yang, "Discriminative locality alignment," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2008, pp. 725–738.
[20] L. Duchene and S. Leclerq, "An optimal transformation for discriminant and principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 978–983, Nov. 1988.
[21] D. Cai, X. He, J. Han, and H. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.
[22] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 506–513.
[23] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2001, pp. 585–591.
[24] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[25] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[26] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2003, pp. 177–184.
[27] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1–8.
[28] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *Proc. 10th IEEE Int Conf. Comput. Vis.*, 2005, pp. 1208–1213.
[29] K. Simonyan, A. Vedaldi, and A. Zisserman, "Learning local feature descriptors using convex optimisation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1573–1585, Aug. 2014.
[30] L. Liu, M. Yu, and L. Shao, "Local feature binary coding for approximate nearest neighbor search," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 1–12.
[31] M. Yu, L. Liu, and L. Shao, "Structure-preserving binary representations for rgb-d action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015. doi: 10.1109/TPAMI.2015.2499125
[32] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.
[33] L. Liu, C. Shen, L. Wang, A. van den Hengel, and C. Wang, "Encoding high dimensional local features by sparse coding based Fisher vectors," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1143–1151.
[34] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3361–3368.
[35] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 1996.
[36] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.
[37] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality reduction: A comparative review," Tilburg Centre for Creative Comput., Tilburg Univ., Tilburg, The Netherlands, Tech. Rep. 2009-005, 2009.
[38] L.-J. Li and F.-F. Li, "What, where and who? Classifying events by scene and object recognition," in *Proc. IEEE 11th Int Conf. Comput. Vis.*, 2007, pp. 1–8.
[39] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, pp. 2169–2178.
[40] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 413–420.
[41] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1794–1801.
[42] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 846–854.
[43] F. Sadeghi and M. F. Tappen, "Latent pyramidal regions for recognizing scenes," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 228–241.
[44] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.
[45] L. Li, H. Su, Y. Lim, and F. Li, "Object bank: An object-level image representation for high-level visual recognition," *Int. J. Comput. Vis.*, vol. 107, no. 1, pp. 20–39, 2014.
[46] B. Liu, Y. Wang, B. Shen, Y. Zhang, and M. Hebert, "Self-explanatory sparse representation for image classification," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 600–616.